Asian Journal of Electrical Sciences (AJES)
Vol.3.No.1 2015 pp 13-17.
available at: www.goniv.com
Paper Received : 08-03-2015
Paper Accepted: 20-03-2015
Paper Reviewed by: 1. R. Venkatakrishnan 2. R. Marimuthu
Editor : Prof. P.Muthukumar

EFFICIENT ENCRYPTION AND DECRYPTION FOR DATA SECURITY USING SIMON BLOCK CIPHERS

G.SETHURAM RAO[1], S.AAZMI BALKEES[2],R.DIVYA[3],M.GOVARTHINI[4]
Department of ECE, Velammal Institute of Technology, Chennai-601204, India
[1]sethuramvit@gmail.com,[2]aazmisaleem@gmail.com,[3]divyachoty@gmail.com,[4]
govarthinimoorthy@gmail.com

ABSTRACT

Lightweight cryptography is a relatively new field that addresses the security issues for highly resource constrained devices. The increasing interest and dependency on compact, portable devices in the recent past has resulted in the need for more efficient hardware implementations of light weight ciphers. SIMON is a light weight block cipher developed and utilized exclusively for resource constraint platforms. It is a strong alternativeto AES (Advanced Encryption standard) in terms of power consumption and hardware overhead. In this work we have efficiently implemented data encryption and decryption using the low-cost, light-weight Simon blockscipher with minimum hardware complexity.

Keywords- Data Security,
Field-Programmable Gate Arrays (FPGA), SIMON, Slices.

I INTRODUCTION:

Cryptography is the method used to send the data in a secured way so that the specific persons only can read and process it.Technically, it is a process of converting the plain text (ordinary text) into cipher text (encryption) and again to get the original text back (decryption).Its main aim is to provide data with confidentiality,integrity and non-repudiation. The process of encryption involves Data (plaintext) which is encrypted using encryption algorithm and encryptedkey.The process generates the cipher text and it can be viewed in original form when it is decrypted.Symmetric and asymmetric key are the two types of keys.Symmetric key encryption uses the same key whereas asymmetric cryptography uses two different keys.

Asymmetric key also known as public key. Public key is shared with everyone and Private Key is kept secret.Symmetric key is much faster than asymmetric,but sender must exchange the key with the recipient before decrypting it.DES, AES are the most widely used encryption algorithms.Encryption finds vast applications in the field of internet, security assurance in IT systems, defence and communication as they provide security. They also provide the following aspects of security-1) Authentication-to verify the location of message,2) Integrity-proof that the contents of the message has not been changed since it was sent, 3)Non-repudiation-the sender of the message can't deny sending the message.

II LITERATURE SURVEY:

The paper proposed in [1]
Two important families of block ciphers SIMON and SPECK having variety of key sizes and widths. The main aim of this development is to develop the need for flexible,secure and analysable light weight block ciphers.Both having lightweight applications,but SIMON is designed to perform well in hardware and SPECK is to give optimal performances for software. The project in **[2]** provides security and consumes less amount of memory space on resource constrained devices lightweight cryptographic algorithms have been developed.
This paper proposes lightweight cryptography for FPGAs by introducing block cipher independent optimization techniques.HIGHT and Preset cryptographic algorithms less than half the size of the AES implementation without using block RAMs. The system proposed in [3] shows that it is a very promising alternative of AES for

resource-constrained platforms. The proposed implementation can execute all configurations of SIMON, and thus provides a versatile architecture that enables adaptive security using a variable key-size.The implementation results show that the proposed architecture occupies 90 and 32 slices on Spartan-3 and Spartan-6 FPGAs, respectively.

III PROPOSED SYSTEM:

The proposed system block diagram is given in fig 1.

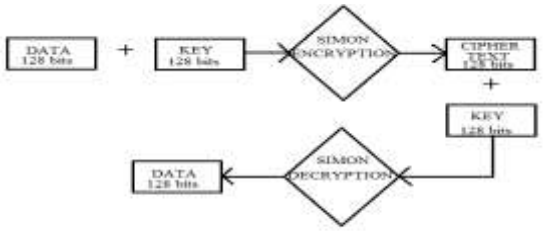

Fig 1

This diagram shows that when 128 bits of data added with 128 bits key, then it undergoes SIMON encryption process and the cipher text of 128 bits is obtained as the output.This 128 bit cipher key is added with the 128 bit key and undergoes a SIMON decryption process.After the decryption,the original data is obtained at the receiver.Here encryption and decryption is done using a new lightweight cryptographic technique called SIMON block ciphers,which is very effective for resource constraint platforms.While AES (Advanced Encryption Standard) is used in number of applications, it occupies larger area and cost limits its involvement in resource constrained applications.And also it requires a separate hardware for both encryption and decryption.Here, we propose a method called SIMON,a recent low cost alternative for resource constrained devices.

A) Simon Encryption:

SIMON encryption is done for 128 bit plain text and 128 bit key to generate 128 bit cipher text in 68 rounds. This SIMON block ciphers supports blocks of sizes 32,48,64,96 and 128 in which each blocks permits three key sizes at a maximum and each family produces 10 algorithms. The following table illustrates the different blocks and key sizes

Block size	**Key sizes**
32	64
48	72,96
64	96,128
96	96,144
128	128,192,256

Table: 1 SIMON parameters

The SIMON block cipher with an n-bit word (2n-bit block) is denotedSIMON2n, where n is required to be 16, 24, 32, 48, or 64. SIMON2n with anm-word (mn-bit) key will be referred to as SIMON2n/mn. For example, SIMON64/128 refers to the version of SIMON using the 64-bit plaintext blocks and 128-bit key.Each object of SIMON uses the Fiestal rule for key generation. The algorithmis designed in a way that is very easy to implement in hardware.SIMON2n encryption for operations onn-bit words as follows:

• Bitwise XOR,
• Bitwise AND, &, and
• left circular shift, Sj, by j bits.

For k€ 2 GF (2) ^{n}, the key-dependent SIMON2n round function is the two-stage
Feistel map R_k:GF (2) ^{n} × GF (2) ^{n}! GF $(2)^{n}$× GF$(2)^{n}$ defined by
$R_k(x, y) = (y \oplus f(x) \oplus k, x)$, where f (x) = (Sx&$s^{8}$x) s^{2}x and k is the round key. The inverse of the roundfunction, used for decryption, is$R^{-1}{}_k(x, y) = (y, x \oplus f(y) \oplus k)$.
The SIMON key schedules take a key and from it generate a sequence of T key
Words k0. . . kT−1, where T is the number of rounds

Blocksize 2n	Key size mn	Word size n	Key words m	Const seq	Rounds T
32	64	16	4	z0	32
48	72	24	3	z0	36
	96		4	z1	36
64	96	32	3	z2	42
	128		4	z3	44
96	96	48	2	z2	52
	144		3	z3	54
128	128	64	2	z2	68
	192		3	z3	69
	256		4	z4	72

Table: 2 Simon parameters

B) Key function:

The key schedule of SIMON is described as a function that will operate on 2, 3 or 4 n-bitword registers, depending on the size of the master key. It performs two rotations to the rightby x >>>3 and x >>> 1 and XOR the results together with a fixed constant c and five constantsequences zijwhich are version-dependent. These constant sequences are obtained by using three 5 X5 matrices over F2 and a linear feedback shift register where the first two are of period 31and the last three are of period 62. The specification makes these constants and

abolishing the sliding properties and circular shift symmetries between the different round keys. And also, they are used to provide cryptographic separation between different versions ofSIMON that have the same block size, but with different key sizes. Fig 2(b) describes the key generation for 128/128 configuration. The value of C constant is equal to (2^n-1) 3, i.e. a string of n-2 ones and two zeroes on the least significant two bits .The valuezijis the i^{th} bit (from most significant to least significant, where i is computed modulo n) of zj, where zj is from Table 3 and j is a parameter of the cipher from table 4.

Table: 3the zj vectors used in the SIMON Key function

Cipher	Block size 2n	Key words m	Key size Mn	Rounds T	Index to Z j
Simon 32/64	32	4	64	32	0
Simon 48/72	48	3	72	36	0
Simon 48/96	48	4	96	36	1
Simon 64/96	64	3	96	42	2
Simon 64/128	64	4	128	44	3
Simon 96/92	96	2	92	52	2
Simon 96/144	96	3	144	54	3
Simon 128/128	128	2	128	68	2
Simon 128/192	128	3	192	69	3
Simon 128/256	128	4	256	72	4

Table: 4 Members of the SIMON family with their parameters

C) Fiestal Round:

The full SIMON round operation consists of three 64-bit shift operators (shift left one, shift left two, and shift left eight) three 64-bit XOR operators and one 64-bit AND operator.The round function performs logic operations on the most significant 64-bits (the upper half block) and it is XOR-ed with the least significant 64-bits (the lower half block) and the 64-bit round key .At the end of each round, the contents of the upper block is transferred to the lower block as the new generated values are written back into the upper block.The key generation function performs logic operations on the most significant 64-bits and the result is XOR-ed with the least significant 64-bits and the 64-bit round constant zi whose values are obtained using table 3 and 4.

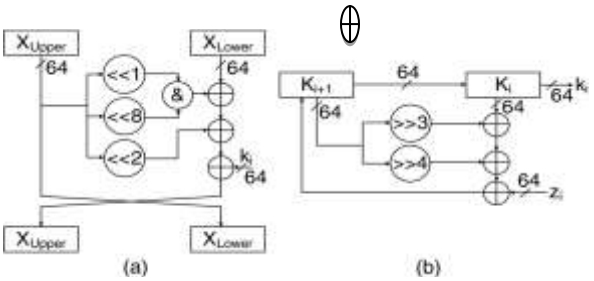

Fig 2. (a) Feistel round and (b) key generation of SIMON for the 128/128 Configuration

J	Zj
0	11111010001001010110000111001101111101000100101011000011100110
1	10001110111110010011000010110101000111011111001001100001010
2	10101111011100000011010010011000101000010001111110010110110011
3	11011011101011000110010111000000100100010100111001101000011111
4	11010001111001101011011000100000010111000011001010010011101111

D) Decryption Process:

At the end of the 68^{th} iteration of the encryption proceess, a 128 bit cipher text is produced at the X upper and X lower blocks of the fiestal round.The obtained cipher text is then given as input to the inverted fiestal architecture along with the round key .The round key is generated from the 128 bit key that was initially used for encrypting the data.The 128 bit original data is deciphered from the cipher text after 68 fiestal rounds.

IV SOFTWARE IMPLEMENTATION AND DESCRIPTION:

The software that is used to do the above operation is Xilinx 12.1 ISE (Integrated synthesis environment).It is primarily used for circuit synthesis and designHDL (Hardware Description Language) languages are used for digital circuit design.The most widely used HDLs are Verilog and VHDL. Verilog HDL is most commonly used in the design, verification, and implementation of digital logic chips at the register transfer level (RTL) of abstraction. We use VHDL (VHSIC hardware description language where VHSIC is an abbreviation for

Very High Speed Integrated Circuit) is a hardware description language which is completely defined by Language Reference Manual (LRM) used in electronic design automation to describe digital and mixed-signal systems such as field-programmable gate arrays and integrated circuits.Simulation and synthesis are the two main kinds of tools which operate on the VHDL language. The Language Reference Manual does not define a simulator, but unambiguously defines what each simulator must do with each part of the language.VHDL does not constrain the user to one style of description. It allows designs to be described using any methodology - top down, bottom up or middle out. VHDL can be used to describe hardware at the gate level or in a more abstract way.

V HARDWARE IMPLEMENTATION

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence "field-programmable". It is shown in Fig 3.

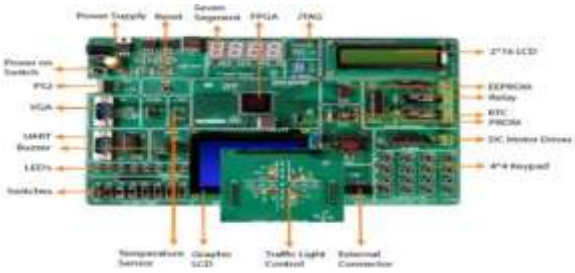

Fig 3

FPGAs contain programmable logic components called "logic blocks", and an order of reconfigurable interconnects that allow the blocks to be "wired together" –like many logic gates that can be inter-wired in different configurations. Logic blocks can be connected to perform complex combinational functions, or simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

The Spartan™-3 families of Field-Programmable Gate Arrays is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications. This Spartan-3 kit , combined with advanced process technology, manufactured with more functionality and bandwidth, setting new standards in the programmable logic industry. Because of their extremely low cost, Spartan-3 FPGAs are ideally suited to a wide range of consumer electronics applications; including broadband access, home networking, display/projection and digital television equipment. The Spartan-3 family is a superior alternative to mask programmed ASICs.

VI DEVICE SPECIFICATIONS :

VII SIMULATION RESULTS:

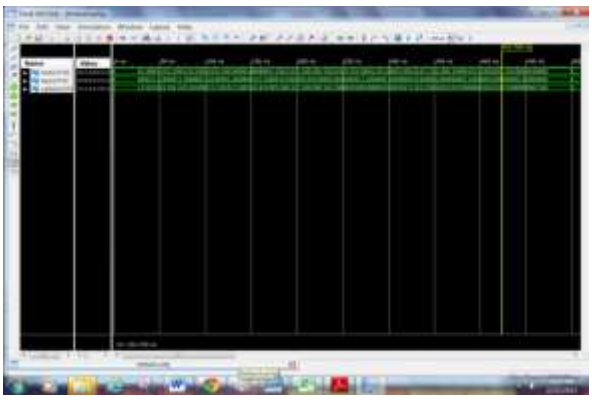

Fig 4. Output for encryption

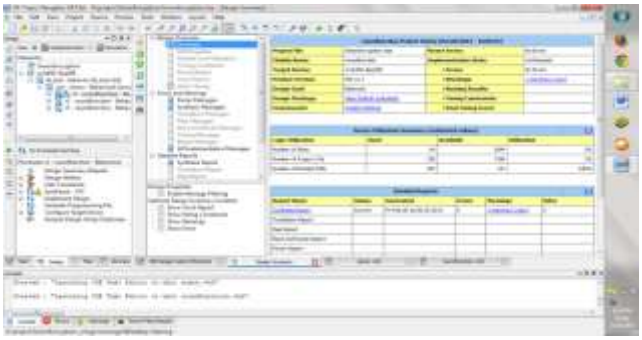

Fig 5. Design summary

VIII PERFORMANCE COMPARISON CHARTS

Type	Design (x,y,z)	Area (slice)
Existing	(1,1,128)	399/3584
Proposed	(1,1,128)	64/3584

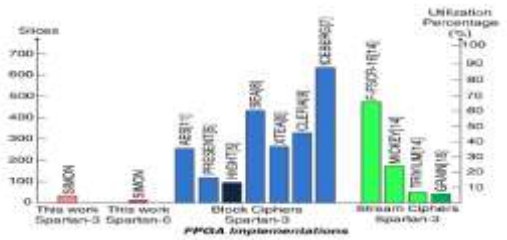

Fig: 6Area utilization chart for SIMON and other block/stream ciphers

IX. ADVANTAGE OF PROPOSED SYSTEM

i. The SIMON lightweight block ciphers significantly reduce the area cost. It can be efficiently implemented in resource-constrained devices.Encryption and decryption can be done using the same hardware.
ii. Smaller in size.
iii. It can be efficiently implemented in resource-constrained devices.
iv. SIMON has a security level equivalent to AES-128 but consumes less area and power.
v. Encryption and decryption can be done using the same hardware.
vi. It is computationally less complex than AES.

X. CONCLUSION:

Bit-serialized implementation of SIMONcost 399 slices on 128 bits for one round on a Spartan-3 FPGA. In this paper we have presented an efficient encryption and decryption for higher end data security made possible by implementing bit parallelism. This hardware architecture consumes only 64 slices on 128 bits for one round on a Spartan-3 FPGA.Itreduces the number of slices, area and power consumption. This significantly decrease the costSIMON claims and have approximately 50% of area reductionover AES for its ASIC implementation, we have shown that with a reduction of 86%, SIMON is an even stronger alternativeto AES for low-cost FPGA applications. We have presented an efficient, low-cost, bit-parallel architecture forthe FPGA implementation of the block cipher SIMON.

XII. REFERENCES:

[1]The Simon and Speck families of Lightweightblock ciphers
[2]Area efficient cryptographic ciphers for resource constrained devices
[3]A Flexible and Compact Hardware Architecture for the SIMON Block Cipher
[4] SIMON Says: Break Area Records of Block Ciphers on FPGAs] L. Henzen and W. Fichtner, "FPGA parallel-pipelined AES-GCM core for 100 g ethernet applications," in Proc. ESSCIRC, 2010, pp. 202–205.
[13] S. Qu, G. Shou, Y. Hu, Z. Guo, and Z.
[5]M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication For RFID systems using the AES algorithm," in Cryptographic Hardware and Embedded Systems CHES 2004, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, 2004, vol. 3156, pp. 357–370.
[6] G. Hembroff and S.Muftic, "Samson: Secure access for medical smart cards over networks," in Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw. 2010, pp. 1–6.
[7] M. Feldhofer, "An authentication protocol in a security layer for RFID Smart tags," in Proc. 12th IEEE Mediterranean Electrotechn. Conf. (MELECON 2004), 2004, pp. 759–762.
[8] R. Laue, O. Kelm, S. Schipp, A. Shoufan, and S. Huss, "Compact
AES-based architecture for symmetric encryption, hash function, and random number generation," in Proc. Int. Conf. Field Programmable Logic Appl. (FPL 2007), 2007, pp. 480–484.
[9] P. Yalla and J. Kaps, "Lightweight cryptography for FPGAs," in Proc. Int. Conf. Reconfigurable Comput. FPGAs (ReConFig '09), 2009, pp. 225–230.
[10] J.-P. Kaps, "Chai-tea, cryptographic hardware implementations of xtea," in Progress in Cryptology-INDOCRYPT 2008, ser. Lecture Notes in Computer Science, D. Chowdhury, V. Rijmen, and A. Das, Eds. Berlin, Germany: Springer-Verlag, 2008, vol. 5365, pp. 363–375.
[11] F.-X. Standaert, G. Piret, G. Rouvroy, and J. J. Quisquater, "FPGA implementations of the iceberg block cipher," in Proc. Int. Conf. Inf. Technol.: Coding Compute. (ITCC 2005), 2005, vol. 1, pp. 556–561.
[12] F. Mace, F.-X. Standaert, and J. J. Quisquater, "FPGA implementation (s) of a scalable encryption algorithm," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 2, pp. 212–216, 2008.
[13] R. Chaves, "Compact CLEFIA implementation on FPGAs," in Embedded Systems Design with FPGAs, P. Athanas, D. Pnevmatikatos, and N. Sklavos, Eds. Berlin, Germany: Springer-Verlag, 2013, pp. 225–243 [Online]. Available: http://dx.doi.org/10.1007/978-1-4614-1362-2_10
[14] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, The SIMON and SPECK families of lightweight block ciphers, 2013.
[15] T. Good and M. Benaissa, "AES on FPGA from the fastest to The smallest," in Cryptographic Hardware and Embedded Systems CHES 2005, ser. Lecture Notes in Computer Science, J. Rao and B. Sunar, Eds. Berlin, Germany: Springer-Verlag, 2005, vol. 3659, pp. 427–440.